

NPS62-85-027

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THE QUEUEING MODEL OF A PACKET SWITCH SUBJECT TO
ROUTING IN COMPUTER NETWORKS

JIN-FU CHANG

SEPTEMBER 1985

Approved for public release, distribution unlimited

Prepared for: Chief of Naval Research
Arlington, VA 22217

FedDocs
D 208.14/2
NPS-62-85-027

710000
L 002.1412
NPS-60-85-021

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral R. H. Shumaker
Superintendent

D. A. Schrady
Provost

The Work reported herein was supported by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

Reproduction of all or part of this report is authorized.

This report was prepared by: 1-1-02

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS62-85-027	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE QUEUEING MODEL OF A PACKET SWITCH SUBJECT TO ROUTING IN COMPUTER NETWORKS		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jin-Fu Chang		8. CONTRACT OR GRANT NUMBER(s) 61152N;RR000-01-NP N0001485WR41005
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943-5100		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Chief of Naval Research Arlington, VA 22217		12. REPORT DATE September 1985
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Approved for public release; distribution unlimited		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Computer Communication Networks, queue, routing, buffer sharing		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The effect of routing on the behavior of a finite queue accepting batch Poisson inputs is studied. Results obtained include state probabilities, blocking probabilities, delay and throughput. Results are also verified by computer simulations. The system considered is a good approximate model of a packet switch in a computer network.		

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

S N 0102- LF- 014- 6601

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ABSTRACT

In a previous paper [1], the steady-state behavior of a finite queue which accepts batch poisson inputs and received service from servers operating in synchronous mode was studied. An analysis was successfully completed via the application of the Residue theorem in complex variables. This work extends the study in [1] to include the effect of routing and buffer sharing. Upon the arrival of a batch each customer determines its route independently according to certain probability distribution. Buffer sharing with minimum allocation studied in [5] is also considered. Results obtained include state probability, blocking probability, delay, and throughput. Validity of analysis has been verified by computer simulations.

1. INTRODUCTION

In a previous paper [1] Chang and Chang study the steady state behavior of a queueing system under the following conditions.

- 1) Customers arrive at the queue in batches according to a Poisson process with mean rate λ batches/sec.
- 2) Each batch carries a random number of customers. The size of each batch is a positive integer-valued random variable which may follow otherwise any arbitrary probability distribution.
- 3) The system provides a capacity of accomodating N customers.
- 4) Customers receive services from m servers in the manner of first-come-first-served. Each customer requires a constant service time of T seconds. The servers operate synchronously in the sense that m customers can be removed form the system constantly every T seconds.
- 5) Upon the arrival of a batch if the remaining space is not enough to accept every customer in the batch then the entire batch will be completely rejected.

The system described above is an approximate model of a packet switch in a computer communication network. Although the problem is combinatorially very complex, it has been successfully solved via the application of the Residue theorem in complex variables. In a separate paper [2], an alternative approach called minislot approximation was introduced to solve the problem.

In a practical system such as computer communication network mentioned above, a packet switch may have several output channels and packets may have their own preferences in selecting an output channel. In order to make the model more practical and the results more valuable, part of the purpose of this study is to extend the work in [1] to include the effect of routing. In

other words, in addition to the above assumptions, should a batch be acceptable each customer of the batch decides from which server to receive service independently of the other customers according to a specific probability distribution. We shall use r_i to denote the probability that a customer will be routed to server i . Clearly, $0 \leq r_i \leq 1$ and $r_1 + \dots + r_m = 1$. This type of routing is usually referred to as random routing in the area of computer communications.

The purpose of buffer sharing among customers routed to different outputchannels is to achieve efficient utilization of buffer. Several papers [3] - [5] have discussed this problem. It is pointed out in [5] that sharing with minimum allocation (SMA) performs better than complete sharing (CS) when traffic is high. In CS the entire buffer is accessible by all customers. In SMA each user has a reserved area which can be accessed only by customers routed to that server. In addition to the reserved region there is still a shared region. This work also tries to examine the effect of SMA on the behavior of a finite queue. For convenience we study the problem in which each server only has a reserved space capable of holding only one customer. Fig. 1 depicts the conceptual model of such a system. We believe the results can be modified to the situation where servers can have reserved areas of different sizes.

The approach using the Residue theorem proposed in [1] will again be used to obtain results such as state probability, blocking probability, average delay, and system throughput. However this approach has to be modified. The modification is necessitated by routing consideration. For example the derivation of state transition probability via the Residue theorem not only depends on the remaining buffer space but also on the set of busy and idle servers. This will be demonstrated in Section 2.

Section 3 of this report discusses the extension of this work to unequal-rate servers. Conclusions are made in Section 4.

Throughout this work ergodicity of the process is assumed so that ensemble-averages can be replaced by time-averages.

2. MAIN RESULTS

Let π_b denote the probability that the system is at state $b = (b_1, \dots, b_m)$ at the beginning of a time slot. In $b = (b_1, \dots, b_m)$ such that $b = b_1 + \dots + b_m$, $b_i (\geq 0)$ represents the number of customers which are to be routed to server i while b denotes the total number of customers in the system. Let $x = (x_1, \dots, x_m)$ such that each x_i is nonnegative integer and $x_1 + \dots + x_m > 0$ be an arriving batch of size $x_1 + \dots + x_m$ in which x_i of them are to be routed to server i . Suppose $N = 10$, $m = 2$, and $b = (1, 2)$. Also suppose three arrivals $(3, 1)$, $(2, 6)$, $(1, 1)$ (according to their order of appearance) have occurred within a slot. The first arrival $(3, 1)$ is acceptable and will lead the system state from $(1, 2)$ to $(4, 3)$. The second arrival $(2, 6)$ must be rejected since $2 + 6 + 4 + 3 = 15 > 10$. The last arrival $(1, 1)$ is acceptable since $1 + 1 + 4 + 3 = 9 < 10$. In other words if $b = (1, 2)$, the arrivals $(3, 1)$, $(2, 6)$, $(1, 1)$ all together result in four and two customers effectively accepted by the system and to be routed to server 1 and 2, respectively. We use $a = (4, 2) = (3, 1) + (1, 1)$ to illustrate such an event and use $p_{a,b}$ to denote the corresponding conditional probability. In general, $\underline{a} = (a_1, \dots, a_m)$ in which each a_i is a nonnegative integer such that $a = a_1 + \dots + a_m > 0$.

Let $n = (n_1, \dots, n_m)$ denote the system state at the beginning of the next slot. Then the steady-state behavior of the system is characterized by

$$(1) \quad \pi_{\underline{n}} = \sum_{(\underline{a}, \underline{b}) \in AB(\underline{n})} \pi_{\underline{b}} p_{\underline{a}, \underline{b}}$$

where

$$(2) \quad AB(\underline{n}) = \{(\underline{a}, \underline{b}) \mid 0 \leq (b_i + 1)^+ < n_i,$$

$$\sum_{i=1}^m (b_i - 1)^+ \leq N - m,$$

$$a_i = n_i - (b_i - 1)^+,$$

$$a + b \leq N,$$

$$\text{and } \sum_{i=1}^m (a_i + b_i - 1)^+ \leq N - m\}$$

In (2),

$$(3) \quad (x)^+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

Physically, $AB(\underline{n})$ represents the set of (a, b) such that the system can transit from \underline{b} to \underline{n} via the acceptance of \underline{a} . In (2), $(b_i - 1)^+$ represents the number of customers (among the initial b_i) remain in the system at the end of the slot. $\sum_{i=1}^m (b_i - 1)^+$ must not be greater than $N - m$ since at most m

customers can be removed from the system per slot. $\sum_{i=1}^m (b_i - 1)^+ = N - m$

occurs when $b = N$, i.e. system is full, at the beginning of a slot and all servers are busy during the slot. $(a_i + b_i - 1)^+$ represents the number of customers which stay in the shared region and to be routed to server i . For example if $b_i = 0$ and $a_i = 1$, then this customer will be placed in the reserved area and there is no customer waiting for server i in the shared area.

Since the shared area is of size $N-m$, we have
$$\sum_{i=1}^m (a_i + b_i - 1)^+ \leq N - m$$

in (2).

$\pi_{\underline{b}}$ in (1) must be solved together with

$$(4) \quad \sum_{\underline{b} \in BB} \pi_{\underline{b}} = 1$$

where

$$(5) \quad BB = \left\{ \underline{b} \mid 0 \leq b_i \leq N - (m - 1), \quad 1 \leq i \leq m, \quad 0 \leq b \leq N \right\}$$

Physically, BB denotes the set of feasible states. In (5), $N - (m - 1)$ represents that the system can have at most $N - (m - 1)$ customers to be routed to server i , $1 \leq i \leq m$.

In order to solve (1), we need to find p_a, b first. Let x_i be a sequence of batch arrivals arranged in the order of their appearances in which $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$. We have explained previously that x_{ij} represents the number of customers in the i th batch of $\{x_i\}$ which select server j to receive their services. Next we use B and I to denote the set of busy and idle servers, respectively within a slot. A customer which arrives within a slot must wait till the beginning of the next slot to receive service even if the server it selects is idle when it arrives. This is because the servers are running synchronously at the same speed. The duration between the arrival of a customer and the beginning of the next slot will be called residual period of this customer. Let R denote the size of the remaining free buffer

including both the shared and reserved region at the beginning of time slot. Let $|B|$ and $|I|$ respectively denote the number of busy and idle servers. Clearly, $R > |I|$ and $R - |I|$ represents the size of free buffer in the shared area.

The approach using the residue theorem proposed in [1] can be used to obtain p_a , p_b . However it has to be modified.

Example 1. Suppose $R = 5$, $m = 3$, $B = \{2\}$, and $I = \{1, 3\}$. This means the shared region can take no more than 3 customers. Thus in $\{x_i\} = \{(4,3,0), (0,3,1), (1,4,2), (5,1,2), (0,0,1), (0,3,0)\}$ only $(0,3,1)$ is acceptable and the rest are rejected. The arrival $(0,0,1)$ is rejected since after the acceptance of $(0,3,1)$ not only the shared region becomes full but also the buffer reserved for server 3 is occupied. Thus the only customer in $(0,0,1)$ which can only be placed in the shared region gets rejected. $\{(0,3,1)\}$ is called an acceptable pattern when $R = 5$, $B = \{2\}$, and $I = \{1, 3\}$. $\{(0,3,1)\}$ further divides $\{x_i\}$ into two subsequences $C_0 = \{(4, 3, 0)\}$ and $C_1 = \{(1, 4, 2), (5,1,3), (0,0,1), (0,3,0)\}$. C_0 and C_1 will be called blocked subsequences of $\{x_i\}$.

In general, we use $\{a_1, \dots, a_n\}$ to denote a general acceptable pattern such that $\underline{a}_k = (a_{k1}, a_{k2}, \dots, a_{km})$ and $a_k = a_{k1} + \dots + a_{km}$. Let v_i and u_i respectively denote the number of unoccupied reserved buffers and the number of unoccupied shared buffers before the acceptance of \underline{a}_{i+1} , $i = 0, 1, \dots, n-1$. For $i = n$, u_n and v_n are defined similarly except after the acceptance of \underline{a}_n . Use R_i to denote the size of the remaining free shared and reserved buffers between the acceptances of \underline{a}_i and \underline{a}_{i+1} . Then any arrival pattern $\{x_i\}$ which contains $\underline{a}_1, \dots, \underline{a}_n$ as a subset may have the following blocked subsequences

$\underline{C}_0, \underline{C}_1, \dots, \underline{C}_n$ such that $c_{ij} \in \underline{C}_i$, $c_{ij} = (c_{ij1} \dots c_{ijm})$ with $c_{ij} = c_{ij1} + \dots + c_{ijm}$ satisfies

(6.a) $c_{0j} > R_0 = R$ or the number of unoccupied reserved buffers selected by customers specified in \underline{C}_{0j} is less than $c_{0j} - v_0$ for $v_0 + 1 \leq c_{0j} \leq u_0 + v_0 = R_0$

(6.b) $c_{ij} > R_j = R = \sum_{k=1}^i a_k$ or the number of unoccupied reserved buffers

selected by customers specified in \underline{C}_{ij} is less than $c_{ij} - v_i$ for $v_i + 1 \leq c_{ij} \leq u_i + v_i = R_i$, $i = 1, \dots, n$.

Define for $i = 1, \dots, n$ and after the acceptance of \underline{a}_i

(7) $B_i = \{j \mid \text{server } j \text{ for which the reserved buffer is occupied by some customer}\}$

(8) $I_i = \{j \mid \text{server } j \text{ idle and its reserved buffer is still empty}\}$

Clearly, $B_i \cup I_i = \{1, 2, \dots, m\}$, $u_i = |I_i|$, and $v_i = |B_i|$. Next, let $B_0 = B$ and $I_0 = I$ where B and I have been defined previously to be the number of busy and idle servers, respectively, at the beginning of a slot. Also define $q_j(x, R_i, B_i, I_i)$ to be the probability that an arriving batch of size x is rejected because the number of servers in I_i selected by customers in the batch is j which is less than $x - v_i$. For example

(9) $q_1(x, R_i, B_i, I_i) = \left(\sum_{\ell \in B_i} r_\ell \right), x \geq v_i + 1$

while

(10) $q_1(x, R_i, B_i, I_i) = \sum_{k \in I_i} \left[\left(\sum_{\ell \in B_i \setminus \{k\}} r_\ell \right)^x - \left(\sum_{\ell \in B_i} r_\ell \right)^x \right]$

$$= \sum_{k \in I_i} \left[\left(\sum_{l \in B_i} r_{lk} \right)^x - q_0(x, R_i, B_i, \{k\}) \right], x \geq v_i + 2$$

In general,

$$(11) \quad q_j(x, R_i, B_i, I_i) = \sum_{(k_1, \dots, k_j)} \left[\sum_{l \in KK_j} \left(\sum_{l \in BU\{k_1, \dots, k_j\}} r_{lk} \right)^x - q_j(x, R_i, B_i, \{k_1, \dots, k_j\}) \right], x \geq v_i + j + 1$$

where

$$(12) \quad KK_j = \{k_1, \dots, k_j\} \mid \text{each } k_l \in I_i \text{ and } k_1, \dots, k_j \text{ distinct}$$

Eq. (11) above can be easily encoded into computer program., Our experiences show the enumeration of $q_j(x, R_i, B_i, I_i)$ does not require too much computer time.

Let β_i denote the probability that an arriving batch is rejected between the acceptance of \underline{a}_i and \underline{a}_{i+1} for $i = 1, 2, \dots, n-1$. β_0 and β_n respectively denote the above probability before the acceptance of \underline{a}_1 and after the acceptance of \underline{a}_n . Then from (6.a) and (6.b)

$$(13) \quad \beta_0 = \bar{g}_{R_0} + \sum_{x=v_0+1}^{R_0} g_x \left[\sum_{j=0}^{x-v_0-1} g_j(x, R_0, B_0, I_0) \right]$$

where g_x denotes the probability that a batch contains a total of x customers and

$$(14) \quad \bar{g}_x = g_{x+1} + g_{x+2} + \dots$$

represents the probability that a batch contains more than x customers.

Next

$$(15) \quad \beta_1 = \bar{g}_{R_1} + \sum_{x=v_1+1}^{R_1} g_x \left[\sum_{j=0}^{x-v_1-1} q_j(x, R_1, B_1, I_1) \right]$$

In general

$$(16) \quad \beta_i = \bar{g}_{R_i} + \sum_{x=v_i+1}^{R_i} g_x \left[\sum_{j=0}^{x-v_i-1} q_j(x, R_i, B_i, I_i) \right]$$

Let N_a be the random variable representing the number of batches arrive in a slot and let AP stand for acceptance pattern then

$$(17) \quad P[AP = \{\underline{a}_1, \underline{a}_2, \dots, \underline{a}_n\} \mid N_a = k] = \begin{cases} \prod_{i=1}^n [g_{a_i} \binom{a_1}{a_{i1}} \dots \binom{a_1 + \sum_{j=1}^{m-1} a_{ij}}{a_{im}} \prod_{j=1}^m (r_j)^{a_{ij}} \sum_{j_0 + \dots + j_n = k - a_n} B_0^{j_0} B_1^{j_1} \dots B_n^{j_n}], & \text{if } k \geq a_n \\ 0, & \text{if } k < 0 \end{cases}$$

where $a_1 + \dots + a_n \leq R$. Via the approach using the Residue theorem [1] we obtain

$$(18) \quad P[AP = \{\underline{a}_1, \underline{a}_2, \dots, \underline{a}_n\} \mid N_a = k]$$

$$= \text{CAP} \frac{1}{2\pi i} \oint |z| = c \frac{z^k}{\prod_{j=0}^n (z - \beta_j)} dz$$

where $c > \beta_0, \beta_1, \dots, \beta_n$ and

$$(19) \quad \text{CAP} = \prod_{i=1}^n \left[g_{a_i} \left(\frac{a_i}{a_{i1}} \right) \dots \left(\frac{a_i}{a_{im}} \right)^{\sum_{j=1}^{m-1} a_{ij}} \prod_{j=1}^m (r_j)^{a_{ij}} \right]$$

(throughout this report the i associated with $2\pi i$ denote $\sqrt{-1}$, otherwise it can be used as a subscript or superscript.) Finally,

$$(20) \quad P[AP = \{\underline{a}_1, \underline{a}_2, \dots, \underline{a}_n\}]$$

$$= \sum_{k=0}^{\infty} P[AP = \{a_1, \dots, a_n\} \mid N_a = k] f_k$$

$$= \frac{\text{CAP}}{2\pi i} \oint |z| = c \frac{F(z)}{\prod_{j=0}^n (z - \beta_j)} dz$$

where $f_k = P[N_a = k]$ and $F(z) = f_0 + f_1 z + f_2 z^2 + \dots$ is the probability generating function of f_k . The derivation of (20) is similar to that in [1] where routing is not considered.

Define

$$(21) \quad \begin{aligned} \Psi_a &= \Psi(a_1 a_2 \dots a_m) \\ &= \{ AP = \{\underline{e}_1, \dots, \underline{e}_n\} \mid e_i = (e_{i1}, e_{i1}, \dots, e_{im}), \end{aligned}$$

$$\sum_{k=1}^n e_{kj} = a_j, \quad 1 \leq j \leq m, \quad 1 \leq n \leq R \}$$

We use $|\psi_a|$ to denote the number of APs contained in ψ_a . Similar to [1], ψ_a can be obtained recursively as follows. If $AP = \{\underline{e}_1, \dots, \underline{e}_n\}$ then define

$$(22) \quad AP * \underline{e}_{n+1} = \{\underline{e}_1, \dots, \underline{e}_n, \underline{e}_{n+1}\}$$

Let $\Psi = \{AP_1, \dots, AP_\ell\}$ where ℓ is an arbitrary positive integer and e an arbitrary acceptable batch, then define

$$(23) \quad \Psi * \underline{e} = \{AP_1 * \underline{e}, \dots, AP_\ell * \underline{e}\}$$

The definitions of (22) and (23) are taken directly from [1].

Example 2

Suppose $m = 2$

$$\Psi_{(1,0)} = \{ \{(1,0)\} \}$$

$$\Psi_{(0,1)} = \{ \{(0,1)\} \}$$

$$\Psi_{(2,0)} = \{ \{(2,0)\} , \{(1,0), (1,0)\} \}$$

$$= \{ \Psi_{(0,0)} * (2,0) \} \quad \{ \Psi_{(0,1)} * (1,0) \}$$

$$\Psi_{(1,1)} = \{ \{(1,1)\} , \{(1,0), (0,1)\} , \{(0,1), (1,0)\} \}$$

$$= \{ \Psi_{(0,0)} * (1,1) \} \quad \{ \Psi_{(1,0)} * (0,1) \} \quad \{ \Psi_{(0,1)} * (1,0) \}$$

$$\Psi_{(2,1)} = \{ \{(2,1)\} \} \quad \{ \{(2,0), (0,1)\} , \{(1,0), (1,0), (0,1)\} \}$$

$$\{ \{(1,1), (1,0)\} , \{(1,0), (0,1), (1,0)\} , \{(0,1), (1,0), (1,0)\} \}$$

$$\{ \{(0,1), (2,0)\} \} \quad \{ \{(1,0), (1,1)\} \}$$

$$= \{ \Psi_{(0,0)} * (2,1) \} \cup \{ \Psi_{(2,0)} * (0,1) \} \cup \{ \Psi_{(1,1)} * (1,0) \}$$

$$\cup \{ \Psi_{(0,1)} * (2,0) \} \cup \{ \Psi_{(1,0)} * (1,1) \}$$

In general, we have

$$(24) \quad \Psi_{\underline{a}} = \Psi(a_1, \dots, a_m)$$

$$= \bigcup_{(k_1, \dots, k_m) \in \Phi(a_1, \dots, a_m)} \{\psi(a_1 - k_1, a_2 - k_2, \dots, a_m - k_m) * (k_1, \dots, k_m)\}$$

where

$$(25) \quad \Phi(a_1, \dots, a_m) = \{(k_1, \dots, k_m) \mid 0 \leq k_1 \leq a_1, \dots, 0 \leq k_m \leq a_m, \text{ but } (k_1, \dots, k_m) \neq (0, \dots, 0)\}$$

Now let us find $P[\Psi_a]$.

Example 3. Let $m = 2$. From our previous discussion of (13), (15) + (16)

we have

$$(26) \quad P[\Psi_{(0,0)}] = \frac{1}{2\pi i} \int_0 \frac{F(z)}{(z - \beta_{00})} dz$$

$$= \frac{1}{2\pi i} \int_0 \frac{A_{00}(z) F(z)}{B_{00}(z)} dz$$

where $A_{00}(z) = 1$, $B_{00}(z) = z - \beta_{00}$, and β_{00} can be obtained from (13)

assuming R_0 , B_0 and I_0 are known.

If $(a_1, a_2) = (1, 0)$ we have

$$(27) \quad P[\Psi_{(1,0)}] = \frac{1}{2\pi i} \int_0 \frac{g_1 r_1 F(z)}{(z - \beta_{00})(z - \beta_{10})} dz$$

$$= \frac{1}{2\pi i} \int_0 \frac{A_{10}(z) F(z)}{B_{10}(z)} dz$$

where $A_{10}(z) = g_1 r_1$, $B_{10}(z) = (z - \beta_{00})(z - \beta_{10})$. In (27), β_{00} and β_{10} can be obtained from (13) and (15) under the assumption that R_0, B_0, I_0 are known and $\{(1,0)\}$ is only acceptable pattern.

Similarly if $(a_1, a_2) = (0,1)$ we have

$$(28) \quad P[\Psi_{(0,1)}] = \frac{1}{2\pi i} \oint \frac{A_{01}(z) F(z)}{B_{01}(z)} dz$$

where $A_{01}(z) = g_1 r_1$, $B_{01}(z) = (z - \beta_{00})(z - \beta_{01})$. In (28), β_{00} and β_{01} can be obtained from (13) and (15) assuming R_0, B_0, I_0 are known and $\{(0,1)\}$ is the only acceptable pattern. If $(a_1, a_2) = (2,0)$, since $\Psi(2,0) = \{\{(2,0)\} \cup \{\Psi(1,0) * (1,0)\}\}$ we have

$$(29) \quad P[\Psi(2,0)] = P[AP = \{(2,0)\}] + P[AP = \{(1,0), (1,0)\}]$$

$$= \frac{1}{2\pi i} \oint \frac{A_{20}(z) F(z)}{B_{20}(z)} dz$$

where

$$A_{20}(z) = g_2 r_1^2 A_{00}(z) \frac{B_{20}(z)}{B_{00}(z)(z - \beta_{00})} + \frac{g_1 r_1 A_{10}(z) B_{20}(z)}{B_{10}(z)(z - \beta_{20})}$$

$$= \sum_{(k_1, k_2) \in \Phi_{(2,0)}} g_{k_1 + k_2} r_1^{k_1} r_2^{k_2} \frac{A_{2-k_1,0} B_{20}(z)}{B_{2-k_1,0}(z)(z - \beta_{20})}$$

$$B_{20}(z) = (z - \beta_{00})(z - \beta_{10})(z - \beta_{20})$$

$$= \prod_{i=0}^2 \prod_{j=0}^0 (z - \beta_{ij})$$

Suppose R_0 , B_0 , and I_0 are given, β_{20} can be obtained either from β_1 of (15) by assuming $AP = \{(2,0)\}$ or from β_2 of (16) assuming $AP = \{(1,0), (1,0)\}$.

In general, for $m = 2$, we have

$$(30) \quad P[\Psi_{(a_1, a_2)}] = \frac{1}{2\pi i} \oint \frac{A_{a_1, a_2}(z) F(z)}{B_{a_1, a_2}(z)} dz$$

where

$$(31) \quad A_{a_1, a_2}(z) = \sum_{(k_1, k_2) \in \Phi} \left[\frac{g_{k_1 + k_2}(k_1 + k_2)}{\phi_{(a_1, a_2)}(k_1)} r_1^{k_1} r_2^{k_2} A_{(a_1 - k_1, a_2 - k_2)}(z) \right. \\ \left. \cdot \frac{B_{(a_1, a_2)}(z)}{B_{(a_1 - k_1, a_2 - k_2)}(z) (z - \beta_{a_1 a_2})} \right]$$

$$(32) \quad B_{(a_1, a_2)}(z) = \prod_{(k_1, k_2) \in \Phi} \frac{(z - \beta_{k_1, k_2})}{BTA(a_1, a_2)}$$

In (32), $BTA(a_1, a_2) = \{\beta_{k_1, k_2} \mid 0 \leq k_1 \leq a_1, 0 \leq k_2 \leq a_1, \text{ and all } \beta_{k_1, k_2} \text{ must have distinct values}\}$. In other words, there could be several combinations of k_1 and k_2 so that their values of β are identical. In this

case, we select only one of them as representative. This can be explained as follows.

In the process of combining all the related rationals to reach

$A_{a_1, a_2}(z)/B_{a_1, a_2}(z)$, it could happen that $(k_1, k_2) \neq (k'_1, k'_2)$ but

$\beta_{k_1, k_2} = \beta'_{k'_1, k'_2}$. In this case we only let the factor $(z - \beta_{k_1, k_2})$ appear

once in $B_{a_1, a_2}(z)$. Thus, $BTA(a_1, a_2)$ used in (32) and defined in the line

following (32) is for this purpose. Notice that $\Phi_{(a_1, a_2)}$ has been defined in

(25). In practice, the value of β_{k_1, k_2} which appears in $B_{a_1, a_2}(z)$ can be

obtained from β_1 of (15) by assuming R_0, B_0, I_0 are given and $AP = \{(k_1, k_2)\}$.

Finally.

$$\begin{aligned} (33) \quad p_{\underline{a}, \underline{b}} &= P(a_1, a_2), \underline{b} \\ &= P[\Psi(a_1, a_2) \mid R_0 = N \neq b, B_0 = B, I_0 = I] \end{aligned}$$

Throughout this example, all the integrals are assumed to be carried out along a circle whose radius is bigger than any of the poles.

$$\begin{aligned}
(34) \quad P[\Psi_a] &= P[\Psi(a_1, \dots, a_m)] \\
&= \frac{1}{2\pi i} \oint \sum_{\underline{k} \in \Phi_a} g_{k_1 + \dots + k_m} \binom{k_1 + \dots + k_m}{k_1} \binom{k_2 + \dots + k_m}{k_2} \dots \binom{k_m}{k_m} \\
&\quad \frac{r_1^{k_1} r_2^{k_2} \dots r_m^{k_m} A_{a_1 - k_1, \dots, a_m - k_m}(z) F(z)}{B_{a_1 - k_1, \dots, a_m - k_m}(z) (z - \beta_{a_1 a_2 \dots a_m})} dz \\
&= \frac{1}{2\pi i} \oint \frac{A_{a_1, \dots, a_m}(z) F(z)}{B_{a_1, \dots, a_m}(z)} dz
\end{aligned}$$

where $\underline{k} = (k_1, \dots, k_m)$ and

$$(35) \quad A_{a_1, \dots, a_m}(z) = \sum_{\underline{k} \in \Psi_a} g_{k_1 + \dots + k_m} \binom{k_1 + \dots + k_m}{k_1} \binom{k_2 + \dots + k_m}{k_2} \dots \binom{k_m}{k_m}.$$

$$\frac{r_1^{k_1} \dots r_m^{k_m} A_{a_1 - k_1, \dots, a_m - k_m}(z) \cdot B_{a_1, \dots, a_m}(z)}{[B_{a_1 - k_1, \dots, a_m - k_m}(z) (z - \beta_{a_1, \dots, a_m})]}$$

$$(36) \quad B_{a_1, \dots, a_m}(z) = \prod_{\beta_{k_1, \dots, k_m} \in BTA(a_1, \dots, a_m)} (z - \beta_{k_1, \dots, k_m})$$

In (36),

$$(37) \quad \text{BTA } (a_1, \dots, a_m) = \{\beta_{k_1}, \dots, \beta_{k_m} \mid 0 \leq k_i \leq a_i \text{ and all } \beta_{k_1} \dots \beta_{k_m} \text{ must have distinct values}\}$$

Also, $\beta_{k_1}, \dots, \beta_{k_m}$ can be obtained from (15) by assuming R_0, B_0, I_0 are known

and $AP = \{(k_1, \dots, k_m)\}$, i.e.,

$$(38) \quad \beta_{k_1}, \dots, \beta_{k_m} = \underline{g}_{R_1} + \sum_{x=v_1+1}^{R_1} g_x \left[\sum_{j=0}^{v_1-1} q_j(x, R_1, B_1, I_1) \right]$$

where

$$(39a) \quad R_1 = R_0 - \sum_{j=1}^m k_j$$

$$(39b) \quad B_1 = \{B_0 \cup \{1 \cdot u(k_1), 2 \cdot u(k_2), \dots, m \cdot u(k_m)\} \cup \{0\}\} - \{0\}$$

$$(39c) \quad I_1 = I_0 \cup \{1 \cdot u(k_1), 2 \cdot u(k_2), \dots, m \cdot u(k_m)\}$$

$$(39d) \quad v_1 = R_1 \wedge |I_1|$$

In (39), $u(x)$ denotes the unit-step function, i.e.,

$$(40) \quad u(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

Notice that $\{0\}$ and $-\{0\}$ in B_1 is to prevent from including 0 and B_1 since servers are numbered from 1 to m .

Finally, $p_{a,b}$ can be evaluated as follows

$$(41) \quad p_{a,b} = P[\Psi_a \mid B_0 = B, I_0 = I, R_0 = N - b]$$

This is because at the beginning of a slot, the initial state \underline{b} is completely specified by the set of busy servers B , the set of idle servers I , and the size of the remaining free buffer space $N-b$. To evaluate the probability that \underline{a} is accepted is equivalent to evaluate Ψ_a based on B , I and $N-b$. Once $p_{a,b}$ is obtained, (1) can be solved. The finiteness of the queue guarantees the existence of the probabilities π_n regardless of the value of λ . The minislot approximation proposed in [2] can also be used to obtain $p_{a,b}$.

Now let $\pi_{\underline{n}}^t$ denote the probability that the system is at state \underline{n} at t

seconds after the beginning of a slot. clearly $\pi_{\underline{n}}^t$ satisfies

$$(42) \quad \pi_{\underline{n}}^t = \sum_{(\underline{a}, \underline{b}) \in \text{ABT}(\underline{n})} \pi_{\underline{b}}^t p_{\underline{a}, \underline{b}}^t$$

subject to

$$(43) \quad \sum_{\underline{n} \in \text{BB}} \pi_{\underline{n}}^t = 1$$

where $\text{ABT}(\underline{n}) = \{(\underline{a}, \underline{b}) \mid b_j \leq n_j \text{ and } a_j = n_j - b_j\}$ and BB has been defined in

(5). Also in (42) $p_{\underline{a}, \underline{b}}^t$ is the probability \underline{a} has been accepted into the system if at the beginning of the slot the system state was \underline{b} . Let 0 denote the beginning of a slot and define

$$(44) \quad \pi_{\underline{n}}^* = \frac{1}{T} \int_0^T \pi_{\underline{n}}^t dt$$

Then

$$(45) \quad \pi_{\underline{n}}^* = \sum_{(\underline{a}, \underline{b}) \in \text{ABT}(\underline{n})} \pi_{\underline{b}}^t p_{\underline{a}, \underline{b}}^*$$

subject to

$$(46) \quad \sum_{\underline{n} \in \text{BB}} \pi_{\underline{n}}^* = 1$$

In (45)

$$(47) \quad p_{\underline{a}, \underline{b}}^* = \frac{1}{T} \int_0^T p_{\underline{a}, \underline{b}}^t dt$$

$$= \frac{1}{2\pi i} \frac{1}{T} \int_0^T \frac{A_{\underline{a}}(z) F^t(z)}{B_{\underline{a}}(z)} dz \quad dt$$

$$\begin{aligned}
&= \frac{1}{2\pi i} \frac{1}{T} \oint \frac{A_a(z)}{B_a(z)} \int_0^T F^t(z) dt \, dz \\
&= \frac{1}{2\pi i} \frac{1}{T} \oint \frac{A_a(z)}{B_a(z)} \int_0^T e^{-\lambda t(1-z)} dt \, dz \\
&= \frac{1}{\lambda T} \frac{1}{2\pi i} \oint \frac{A_{a_1, \dots, a_m}(z) [e^{\lambda T(z+1)} - 1]}{B_{a_1, \dots, a_m}(z) (z+1)} dz \quad \left| \begin{array}{l} R_0=R, B_0=B, I_0=I \end{array} \right.
\end{aligned}$$

The derivation of (47) is similar to the derivation of $p_{a,b}$.

We use B_b^t to denote the probability that a batch arrives at t $(0, T)$ is rejected. Also we use B_b to denote the average blocking probability of a batch. Clearly,

$$(48) \quad B_b = \frac{1}{T} \int_0^T B_b^t dt$$

$$= \sum_{n \in BB} \pi_n^* \left\{ g_{N-n}^* + \sum_{x=v(n)+1}^{R(n)} g_x \left[\sum_{j=0}^{x+v(n)-1} q_j(x, R(n), B(n), I(n)) \right] \right\}$$

where

$$(49a) \quad B(\underline{n}) = \{ \{ 1 \cdot u(n_1), 2 \cdot u(n_2), \dots, m \cdot u(n_m) \} \cup \{0\} \} \approx \{0\}$$

$$(49b) \quad I(\underline{n}) = \{1, \dots, m\} \approx B(\underline{n})$$

$$(49c) \quad R(\underline{n}) = N \approx b$$

$$(49d) \quad v(\underline{n}) = R(\underline{n}) - |I(\underline{n})|$$

Define a test customer to be a randomly selected customer. Clearly, the probability that this test customer is drawn from a batch of size x is $xg_x/$

$(g_1 + 2g_2 + \dots)$. Let B_c^t be the blocking probability of a test customer t seconds after the beginning of a slot and

$$B_c = \frac{1}{T} \int_0^T B_c^t dt,$$

then

$$(50) \quad B_c = \frac{1}{a} \sum_{n=0}^N \pi_n^* \left\{ \sum_{x=N-n+1}^{\infty} xg_x + \sum_{x=v(\underline{n})+1}^{R(\underline{n})} xg_x \cdot \left[\sum_{j=0}^{x-v(\underline{n})-1} q_j(x, R(\underline{n}), B(\underline{n}), I(\underline{n})) \right] \right\}$$

where

$$(51) \quad \alpha = \sum_{x=1}^{\infty} xg_x$$

represents the average batch size.

Let L_q denote the average number of customers waiting in the system, then

$$(52) \quad L_q = \sum_{\underline{b} \in BB} \left[\sum_{\underline{a} \in AA(\underline{b})} \left(\sum_{j=1}^m a_j + (b_j - 1)^+ \right) p_{\underline{a}, \underline{b}}^* \right] \pi_{\underline{b}}$$

where $AA(\underline{b})$ denotes the set of feasible \underline{a} if the system was initially at \underline{b} and can be expressed as follows

$$(53) \quad AA(\underline{b}) = \{ \underline{a} \mid 0 \leq a \leq N - b, 0 \leq \sum_{j=1}^m (a_j + b_j - 1)^+ \leq N - m \}$$

The reason that $\sum_{j=1}^m (a_j + b_j - 1)^+ \leq N - m$ has been given in the explanation

of (2). Next let $L_{q,k}$ denote the average queue length associated with server k , then

$$(54) \quad L_{q,k} = \sum_{\underline{b} \in BB} \left[\sum_{\underline{a} \in AA(k)} (a_k + (b_k - 1)^+) p_{\underline{a}, \underline{b}}^* \right] \pi_{\underline{b}}$$

We use $\lambda_{e,k}$ to denote the effective input rate, expressed in number of customers per sec, to server k , then

$$(55) \quad \lambda_{e,k} = \lambda \sum_{\underline{n} \in BB} \pi_{\underline{n}}^* \left[\sum_{i=1}^{N-n} g_i AR(\underline{n}, i, k) \right]$$

where $AR(\underline{n}, i, k)$ denotes the average number of customers accepted by server k if the batch contains i customers and the system is at state \underline{n} when the batch arrives. In (55), i ranges from 1 to $N - n$ since it is necessary for a batch not to contain more than $N - n + 1$ if that batch is to be accepted. $AR(\underline{n}, i,$

k) can be accepted as follows. First we define an indicator variable $IV(k)$ as follows to indicate whether k is in $I(n)$ of (49.b)

$$(56) \quad IV(k) = \begin{cases} 1, & \text{if } k \in I(n) \\ 0, & \text{elsewhere} \end{cases}$$

Using the notations defined in (49) we obtain

$$(57) \quad AR(n, i, k) = \begin{cases} \sum_{j=1}^i j \binom{i}{j} r_k^j (1 - r_k)^{i-j}, & \text{if } i \leq v(n) + IV(k) \\ v(n) \sum_{j=1}^{v(n) + IV(k)} j \binom{i}{j} r_k^j \{(1 - r_k)^{i-j} - \\ i - v(n) \sum_{x=0}^{i - v(n) + IV(k)} [q_x(i-j, R(\underline{n}), B(\underline{n}), I(\underline{n}))]\} \\ , & \text{if } v(n) + IV(k) \leq i \leq v(n) + |I(n)| \end{cases}$$

This can be explained as follows. When $i \leq v(n) + IV(k)$ then the batch will definitely be accepted. Therefore in this case the average number of

customers accepted is simply $\sum_{j=1}^i j \binom{i}{j} r_k^j (1 - r_k)^{i-j}$. However when

$v(n) + IV(k) \leq i \leq v(n) + |I(n)|$ then the batch of size i could be rejected due to improper routing. The term $\sum q_x(i-j, R(\underline{n}), B(\underline{n}), I(\underline{n}))$ accounts for this effect and must be subtracted.

Once $L_{q,k}$ and $\lambda_{e,k}$ are obtained, the average waiting time of an accepted customer routed to server k , $W_{0,k}$, can be obtained from Little's formula as follows.

$$(58) \quad W_{c,k} = \frac{L_{q,k}}{\lambda_{e,k}}$$

This holds at steady state.

Then the average delay of an accepted customer routed to server k is

$$(59) \quad D_{c,k} = W_{c,k} + T$$

Let D_c denote the average delay of a customer, then

$$(60) \quad D_c = \sum_{k=1}^m D_{c,k} r_k$$

Although the derivation of D_c is done through (58) via Little's formula, the average batch delay D_b has to be obtained in a different manner. This is of course due to the variation in batch size. But more importantly, customers in the same batch can be routed to different servers. This complicates the derivation of D_b .

Let $D_{b,\ell}$ denote the average delay of a batch of size ℓ . Then

$$(61) \quad D_{b,1} = \sum_{(\underline{a}, \underline{b}) \in AB_1} \left\{ \frac{\pi_b}{p_{\underline{a}, \underline{b}}} p_{\underline{a}, \underline{b}}^* \right\} \sum_{k_1 \in KK_1} [a_{k_1} + (b_{k_1} + 1)^+ - 1 + 0.5] r_{k_1} \Bigg\}$$

where

$$(62) \quad AB_1 = \{(\underline{a}, \underline{b}) \mid 0 \leq a + b \leq N - 1, \sum_{j=1}^m (a_j + b_j + 1)^+ \leq N - m\}$$

In (61), the 1 corresponds to 1 slot of service time while the 0.5 represents the average residual period. The residual period of a customer is measured from the instant of its arrival till the beginning of the next slot. Since arrival could occur anywhere within a slot, the average residual period turns out to be 0.5 slots. The KK_1 in (61) denotes the set of servers which guarantees the acceptance of the batch of size 1. Obviously, if there is at least one empty buffer in the shared region then $KK_1 = \{1, 2, \dots, m\}$. If the shared buffer is already full, then KK_1 is simply the set of servers for which their reserved buffers are empty. For $D_{b,2}$ we have

$$(63) \quad D_{b,2} = \sum_{(a,b) \in AB_2} \pi_b p_{a,b}^* \left\{ \sum_{(k_1, k_2) \in KK_2} \max[a_{k_1} + (b_{k_1} - 1)^+ + d_{k_1} + 0.5, \right. \\ \left. a_{k_2} + (b_{k_2} - 1)^+ + d_{k_2} + 0.5] r_{k_1} r_{k_2} \right\}$$

where d_i denotes the number of customers in the batch of size 2 which select server i and

$$(64) \quad AB_2 = \{(a, b) \mid 0 \leq a + b \leq N - 2, \sum_{j=1}^{m'} (a_j + b_j - 1)^+ \leq N - m\}$$

$$(65) \quad KK_2 = \{(k_1, k_2) \mid \text{servers } k_1 \text{ and } k_2 \text{ are chosen by the batch and}$$

$$\sum_{i=1}^m (a_i + b_i + d_i - 1)^+ \leq N - m \}$$

In general, we have

$$(66) D_{b,l} = \sum_{(\underline{a}, \underline{b}) \in AB_l} \pi_{\underline{b}}^* p_{a,b}^* \left\{ \sum_{(k_1, \dots, k_l) \in KK_l} \max [a_{k_1} + (b_{k_1} - 1)^+ + d_{k_1} + 0.5, \dots, a_{k_l} + (b_{k_l} - 1)^+ + d_{k_l} + 0.5] r_{k_1} \dots r_{k_l} \right\}$$

where

$$(67) AB_l = \{(a, b) , 0 \leq a + b \leq N - l, \sum_{j=1}^m (a_j + b_j - 1)^+ \leq N - m\}$$

and,

$$(68) KK_l = \{(k_1, \dots, k_l) \text{ servers } k_1, \dots, k_l \text{ are chosen by the batch and}$$

$$\sum_{i=1}^m (a_i + b_i + d_i - 1)^+ \leq N - m\}$$

Finally, the average batch delay is given by

$$(69) D_b = \frac{1}{1 - B_b} \sum_{l=1}^N g_l D_{b,l}$$

Concerning delay analysis, most papers in the literature concentrate on customer delay, mainly for simplicity. For customer delay, relation between delay and throughput or input rate can be established usually through Little's formula. However, no such relation is guaranteed for batch or group delay. The reasons for the problem studied in this report are variation in batch size and routing effect. Similar phenomenon is also noticed by Whitt [6] and Halfin [7].

The system throughput S is given by

$$(70) \quad S = \sum_{b \in BB} \sum_{k=1}^m u(b_k) \pi_b$$

where BB is defined in (5).

We have carried out extensive numerical calculations based on the results obtained here. Fig. 2-5 are part of them. In these examples we assume $m = 3$, $N = 8$ and $r_1 = r_2 = r_3 = 1/3$. Two batches size distributions are considered in each of these examples. One is the well known geometric distribution with $p = 0.5$ or $\alpha = 2$. The other is specified by $g_1 = g_8 = 0.267$, $g_2 = g_7 = 0.133$, $g_3 = g_6 = 0.067$, and $g_4 = g_5 = 0.033$. Since the latter has a shape looks like a suspension bridge this distribution will be referred as SB for convenience.

In each figure, the curve labeled by G corresponds to geometric distribution while the one labeled by SB corresponds to the suspension bridge distribution mentioned above. In addition to numerical calculations we also carry out simulations to support our analysis. We observe in these figures that the agreement between analysis and simulation is extremely good.

Fig. 2 shows B_0 vs λ . First, we observe that in both G and SB, B_0 is an increasing function of λ . This is intuitively reasonable. As a matter of fact, B_0 approaches 1.0 as λ approaches infinity. The rate of convergence depends on the actual distribution of the size. Second, we observe that SB distribution exhibits higher blocking probability than G. This is explainable. Since the size of a batch in SB is restricted to occur in the range between one and eight $g_1 = g_8 = 0.267$ is considerably higher than the rest. On the contrary, for geometric distribution with $p = 0.5$, $g_1 = 0.5$ and the

distribution is strictly decreasing in batch size. This then implies SB had higher blocking probability. Third the gap between G and SB closes up as λ increases. This is reasonable since as λ becomes big enough the system occupancy is high and most of the arrivals will be rejected regardless of their actual size distribution.

Fig. 3 and 4 show D_C and D_D , respectively, versus λ . For G distribution we observe both D_C and D_D increase as λ increases. This is reasonable. However, for SB, D_C and D_D first decreases then increases as λ increases from zero. This can be explained as follows. As λ is low, almost every arrival can be accepted. Since the batch of size 8 has a good chance of entering the system when λ is small, the overall delay is high. When λ gradually increases, only short batches can enter the system thus the average delay of the accepted batches drops. As λ is high enough, system is full almost all the time thus the delay again is high. Notice that the disagreement between analysis and simulation is less than 5%.

Fig. 5 shows system throughput versus λ . Here we observe G has higher throughput. This phenomenon is consistent with Fig.2.

3. Extensions to Unequal Rate Servers

Although routing has been considered in Section 2, the servers there are still assumed to be operating at the same speed. In practice, servers attached to the queue may run at different speeds. In order to move one step forward in the modeling of a packet switch, the purpose of this section is to extend the results to include servers possible running at different rates.

In this section we assume when a customer is routed to server k it will take the server (or transmitter) Δ_k seconds to complete its service for this customer. Due to the synchronous nature of servers, the behavior of the queue can be described on the basis of T -second slots where T is the greatest common divisor of $\Delta_1, \dots, \Delta_m$. In addition to the random routing considered above we also consider another type of routing called idle-server-first (ISF) routing which is believed to offer better performance. In ISF routing, idle servers or servers with no customer waiting will be considered first upon the arrival of a packet. For both types of routing we have established state-transition equations, obtained state transition probabilities, and derived results such as blocking probabilities, delays, and throughputs, etc. Based on these results we have also carried out extensive numerical calculations. The validity of analysis has also been verified by computer simulations. Figures. 6-9 show part of the numerical results.

4. Conclusions

We have in this work studied the effect of routing and buffer sharing with minimum allocation on the behavior of a finite queue which receives batch Poisson inputs and provides multiple servers running synchronously at the same speed. The main contribution of this work is to obtain analytical results for system state probability, blocking, delay and throughput. The validity of analysis is not only verified by simulation but also supported by intuitive reasonings. The work reported here is an extension of [1].

We have also extended the model to include unequal rate servers.



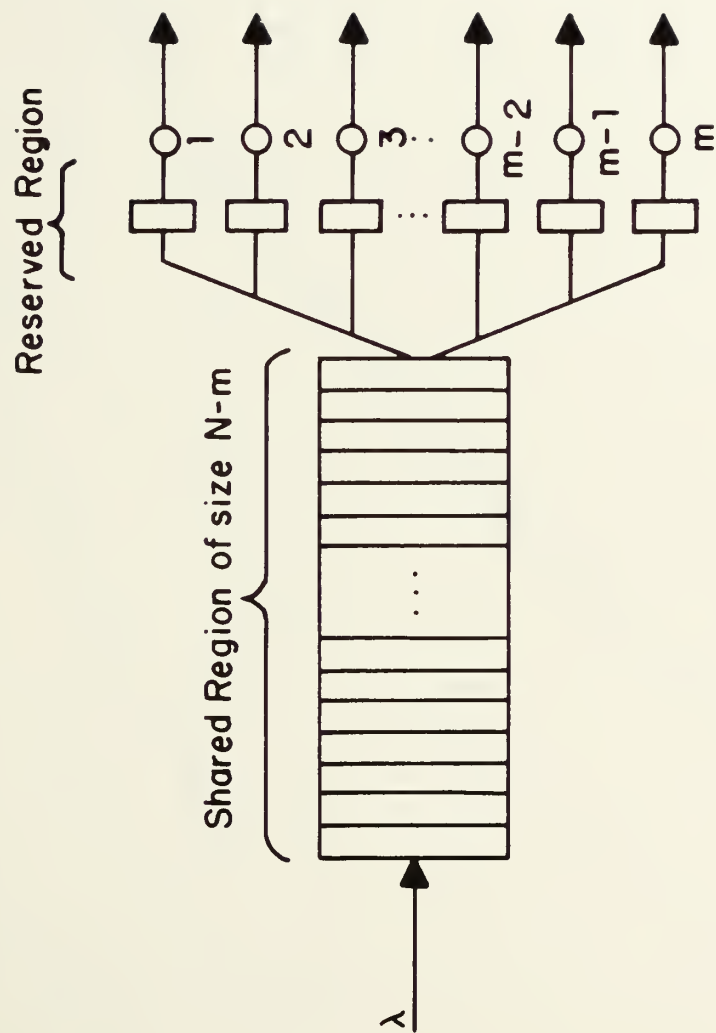


Figure 1.

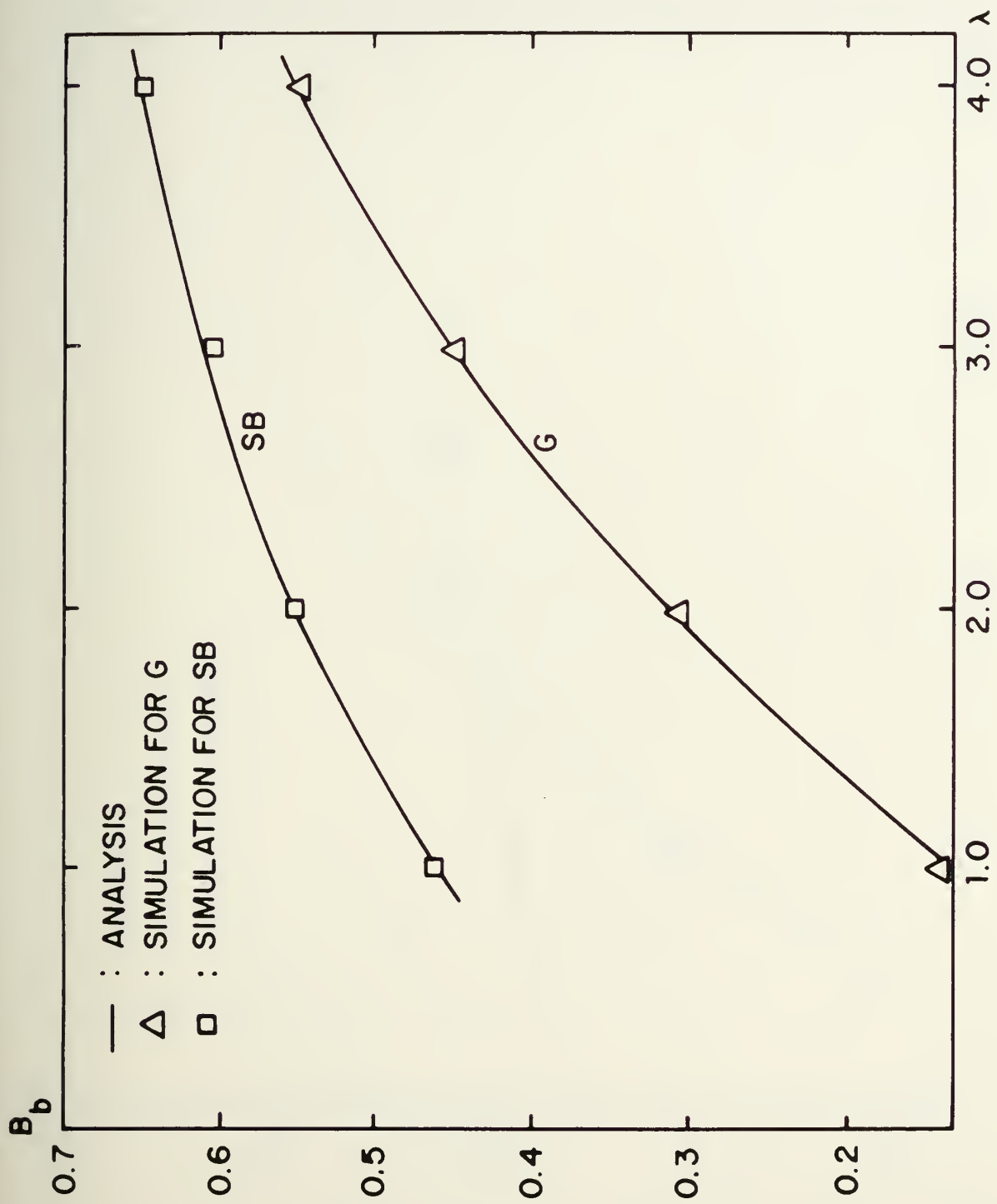
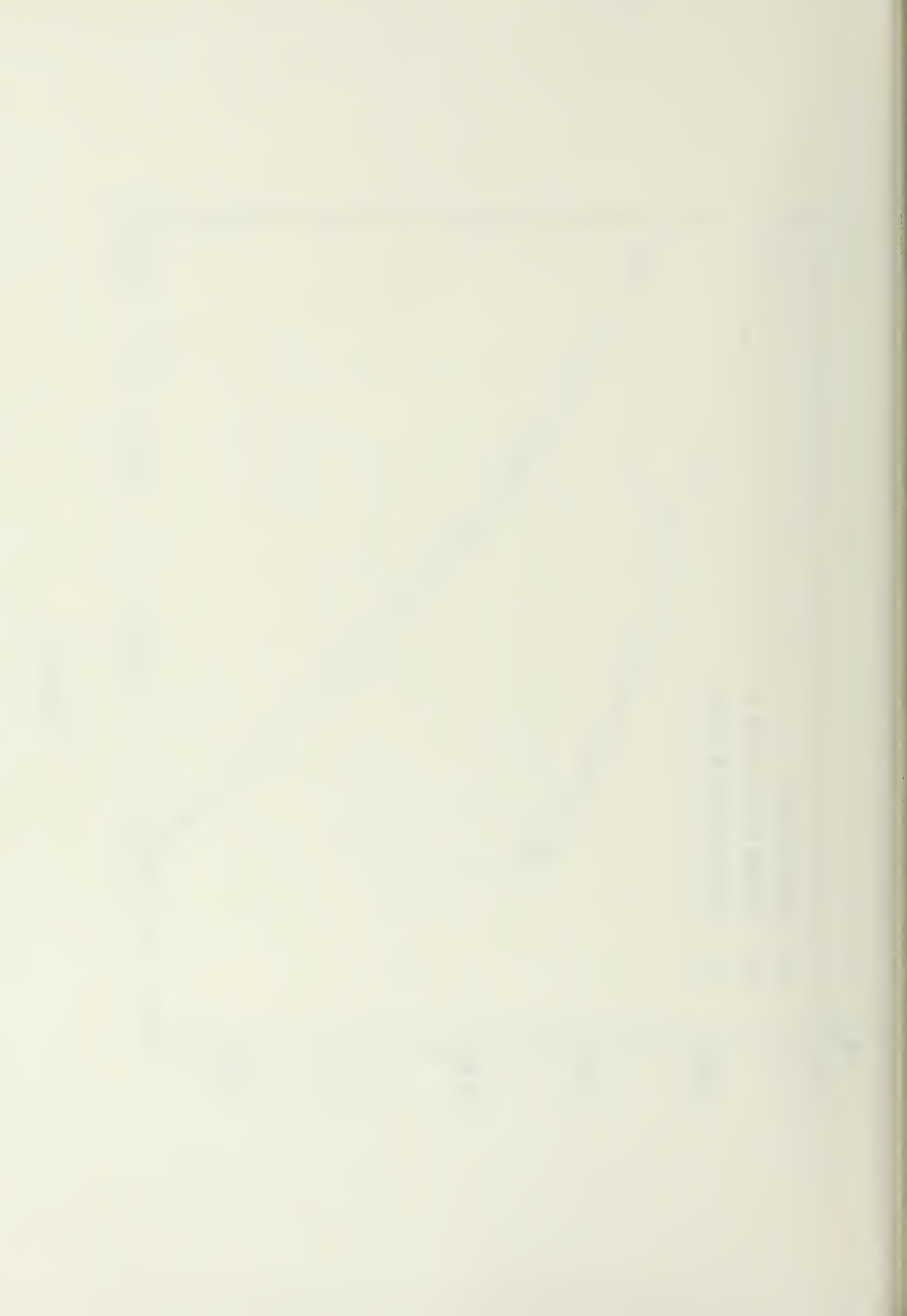


Figure 2.



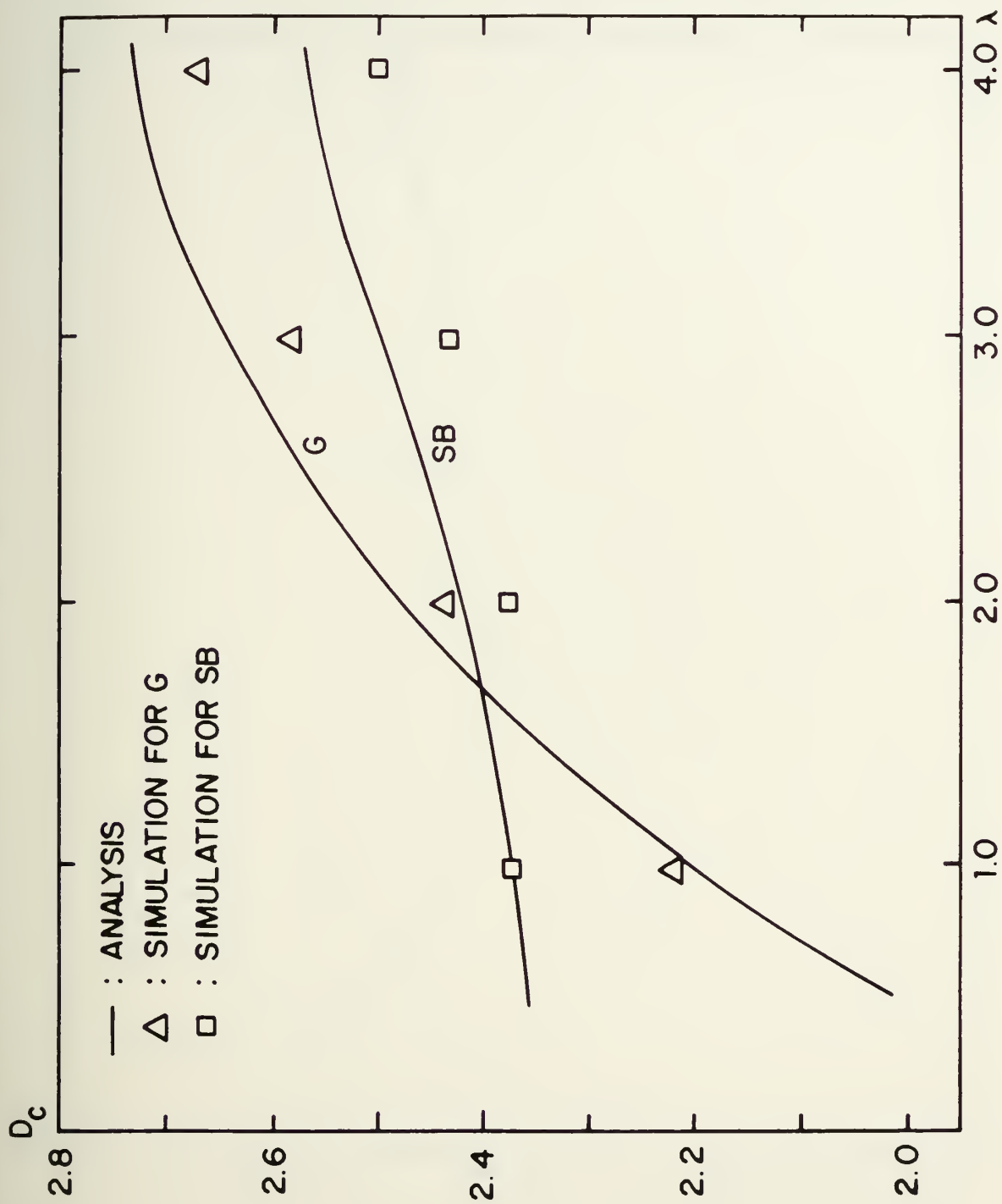
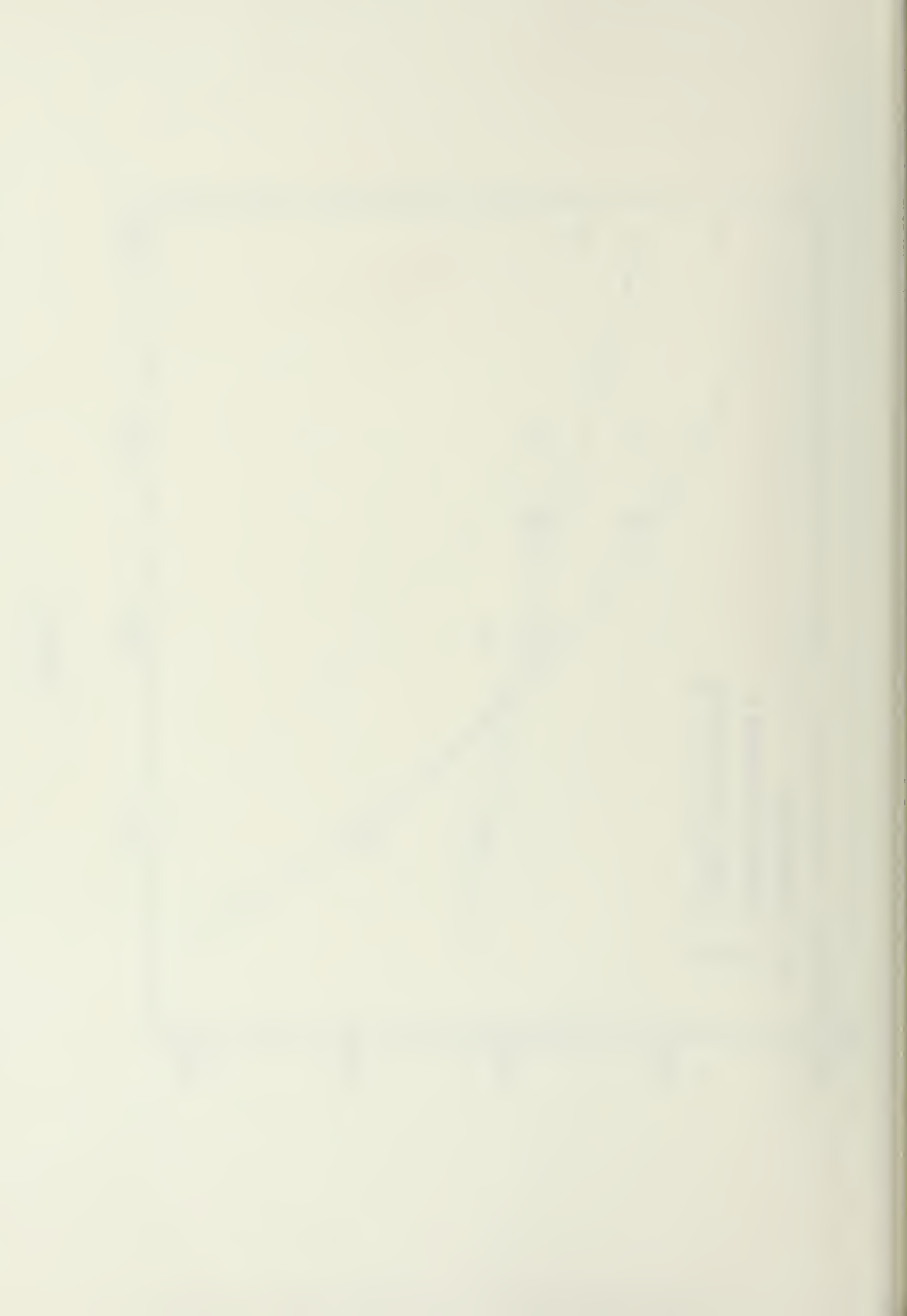


Figure 3.



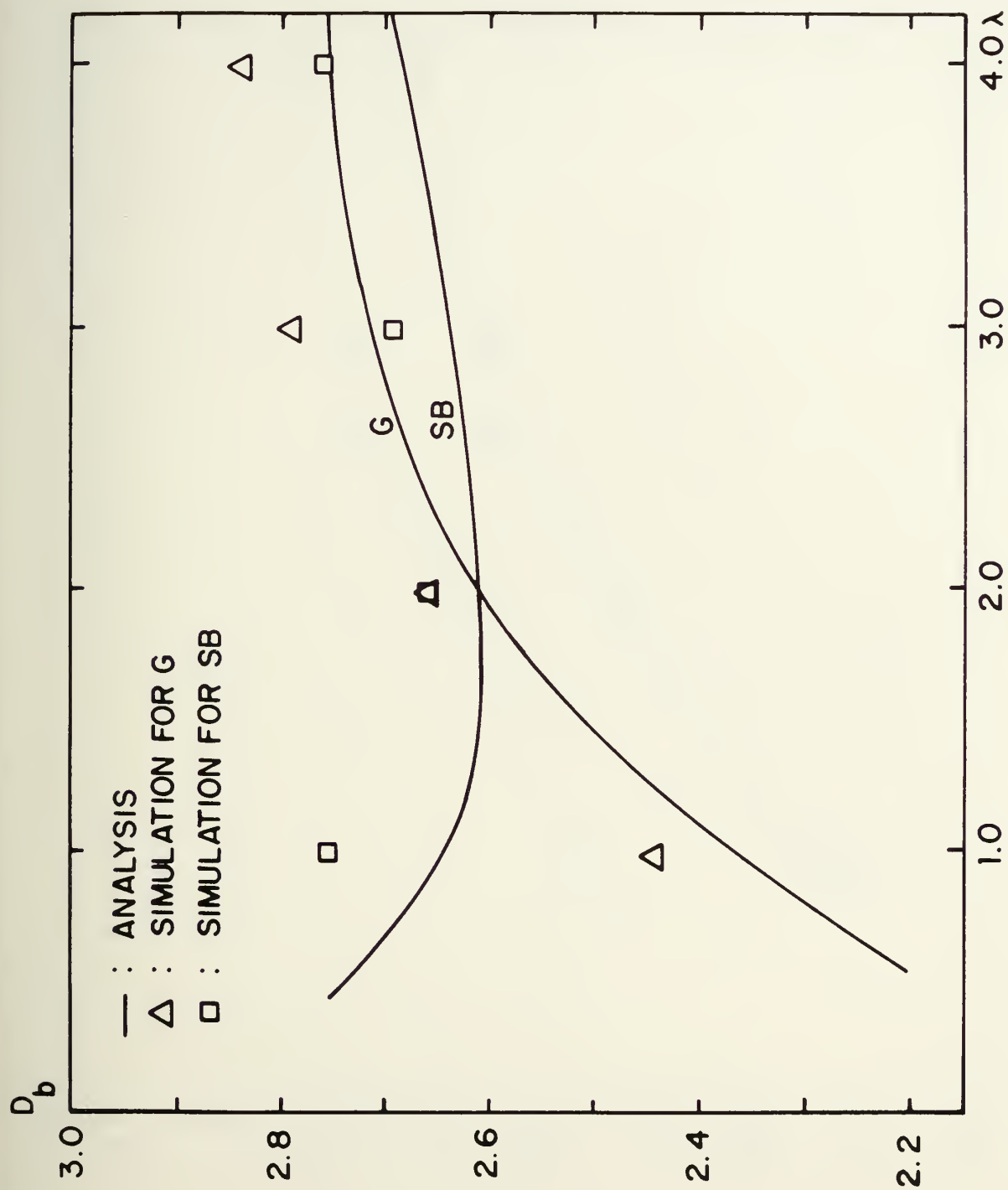
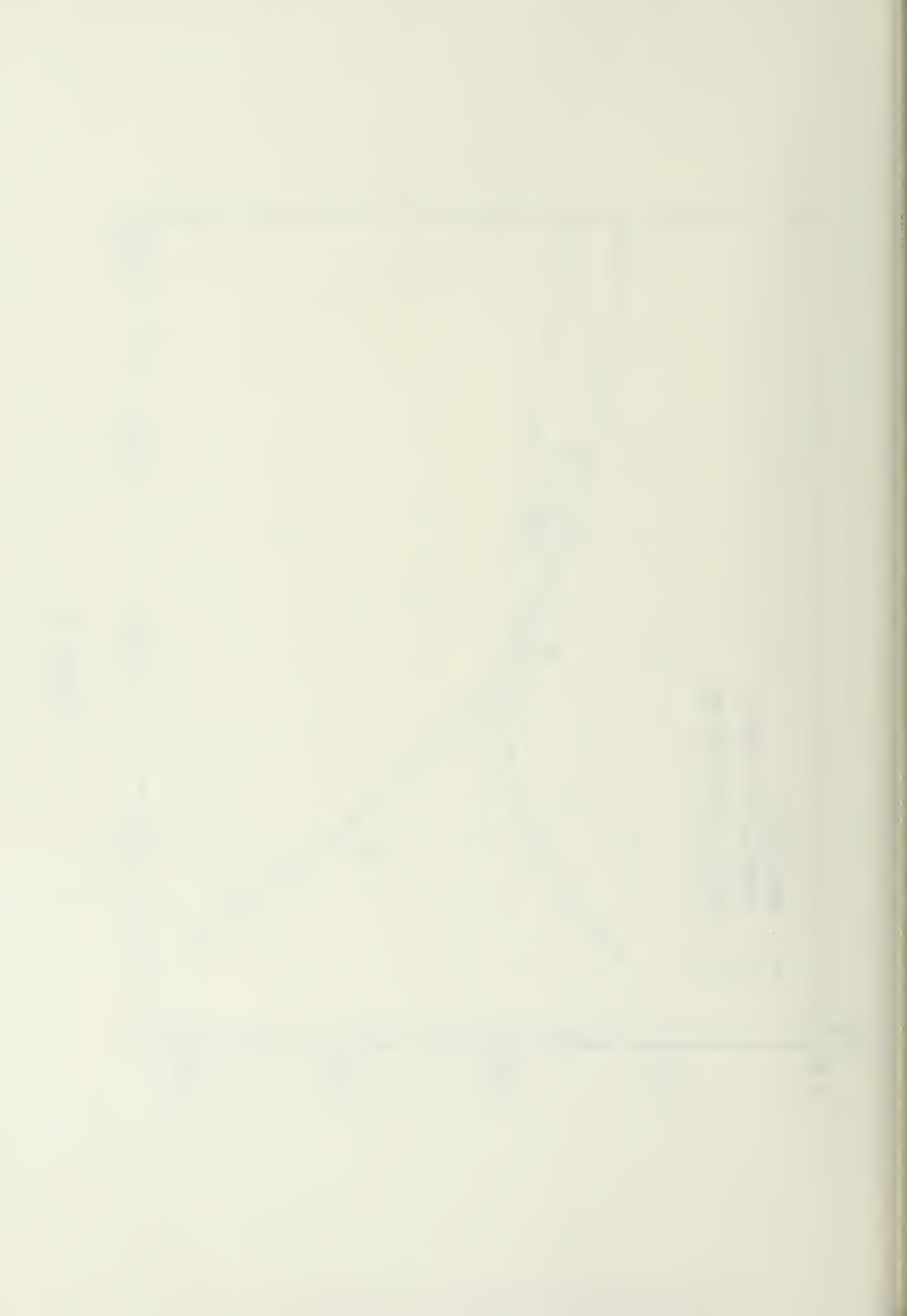


Figure 4.



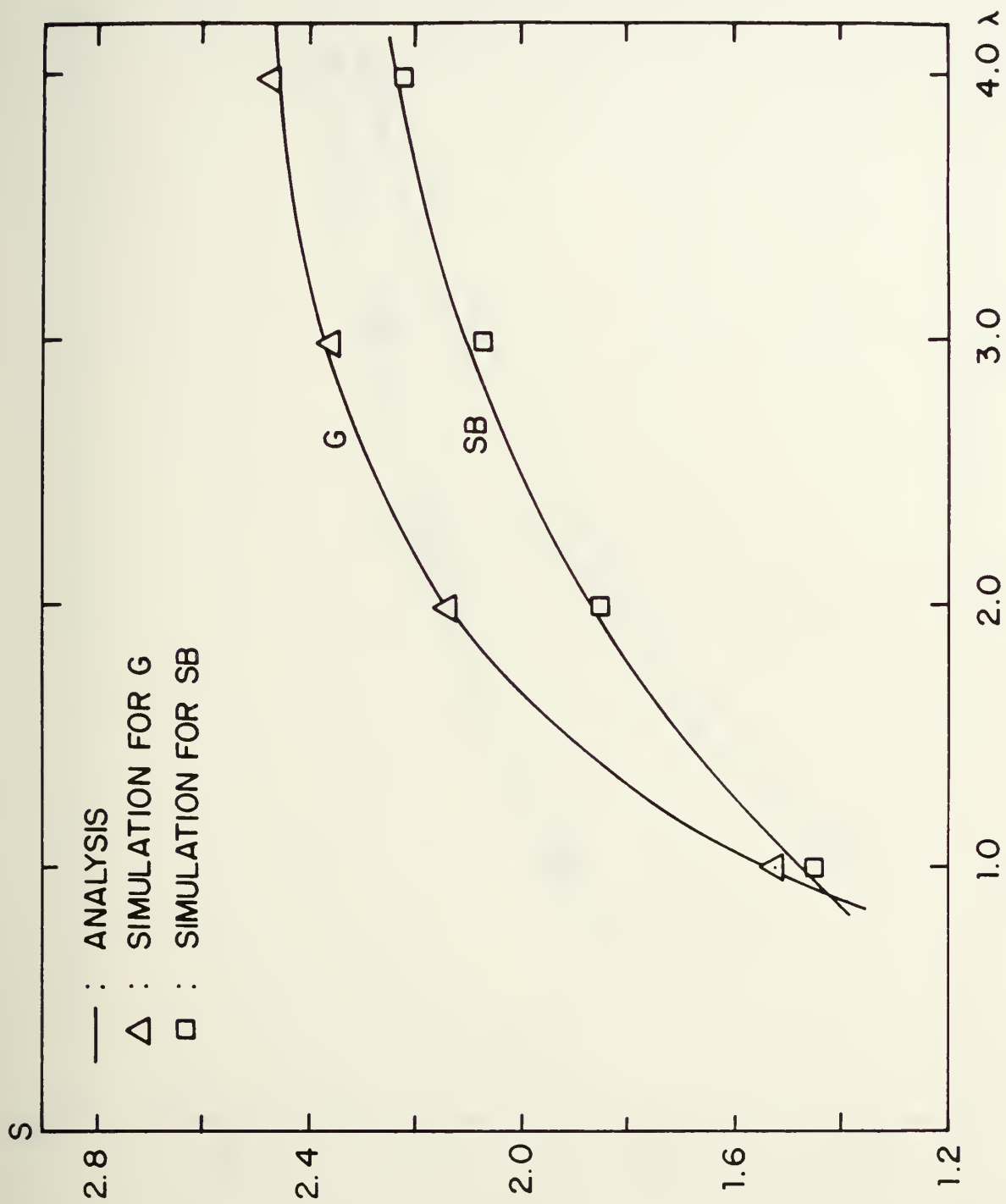


Figure 5.



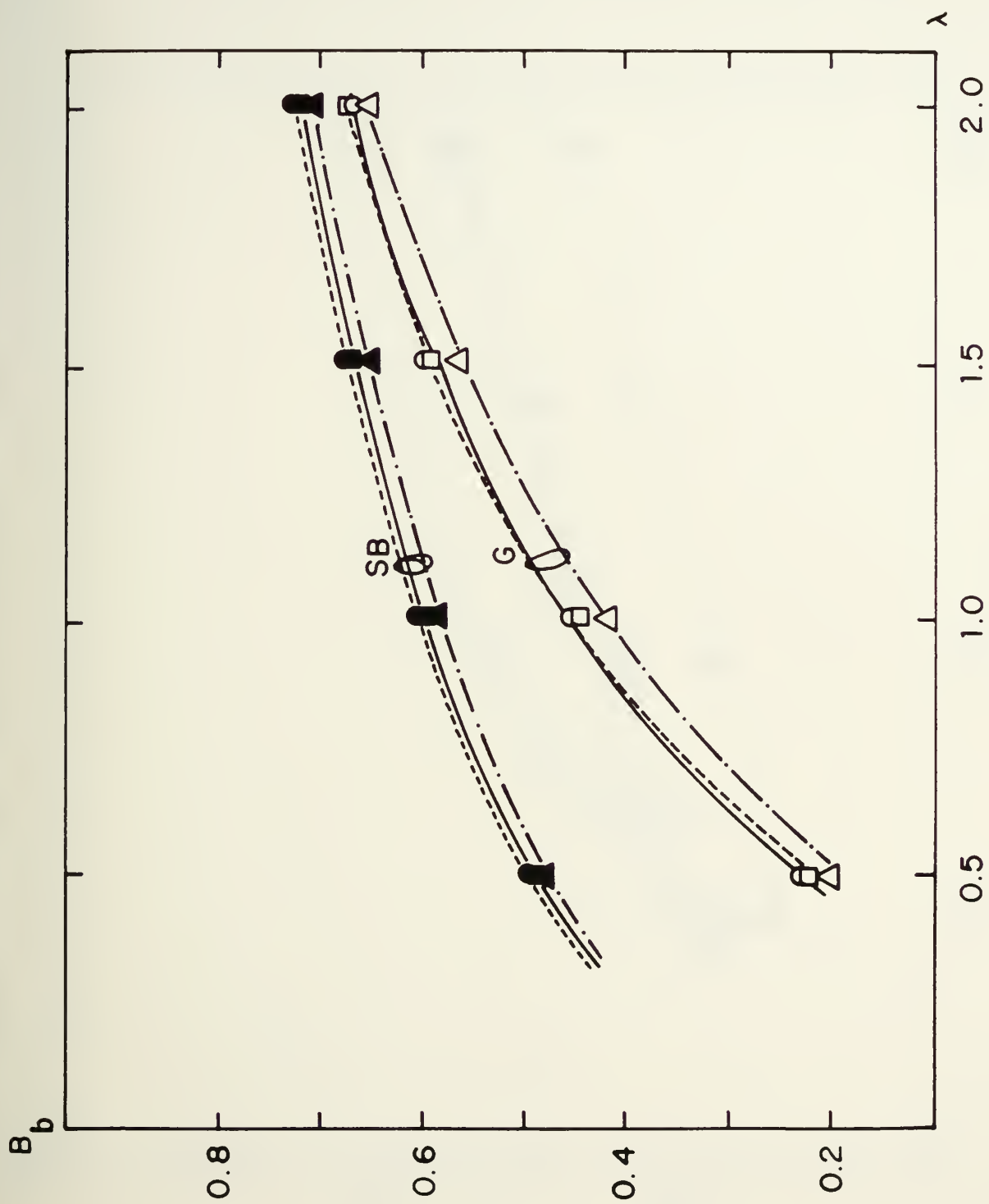
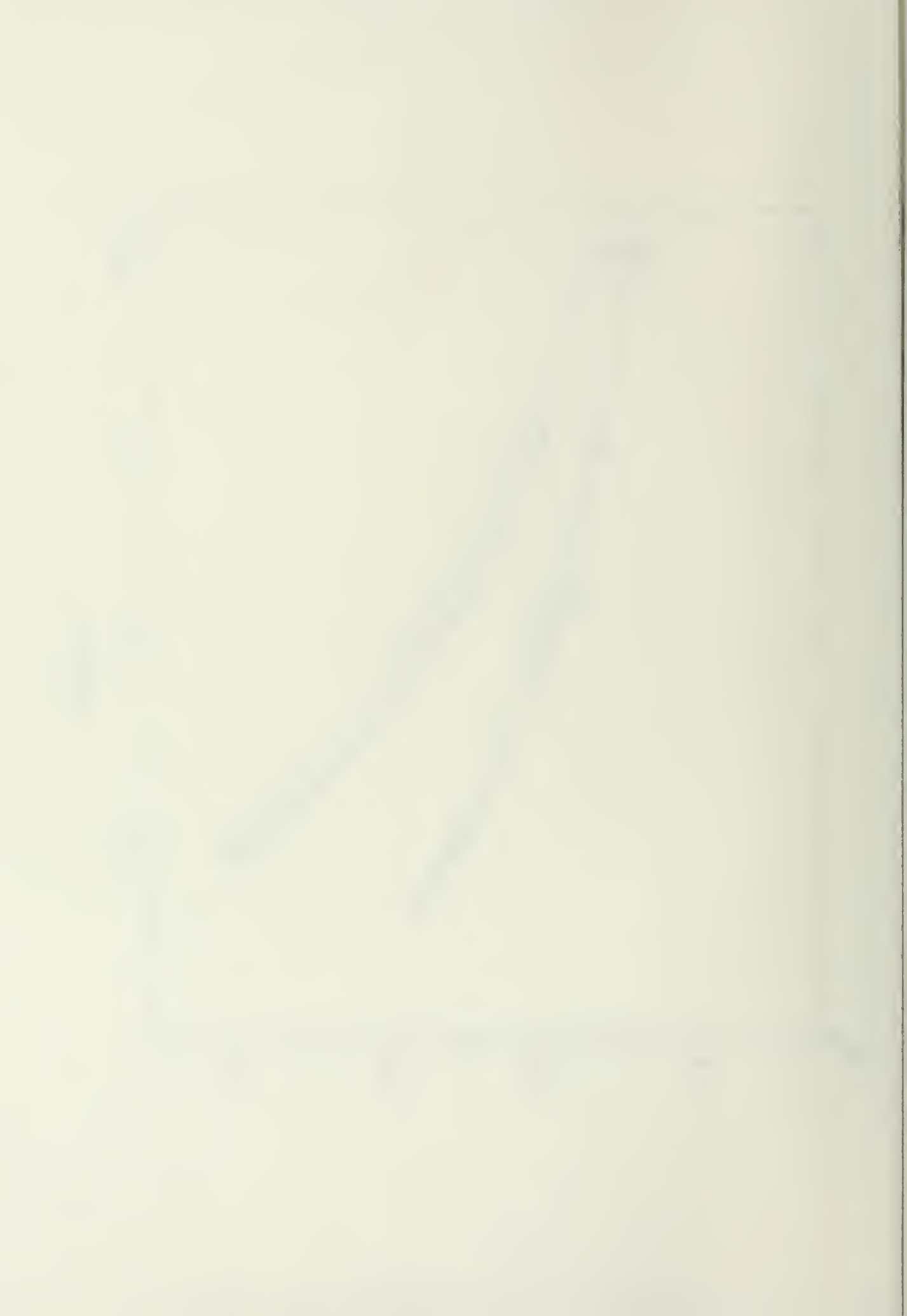


Figure 6.



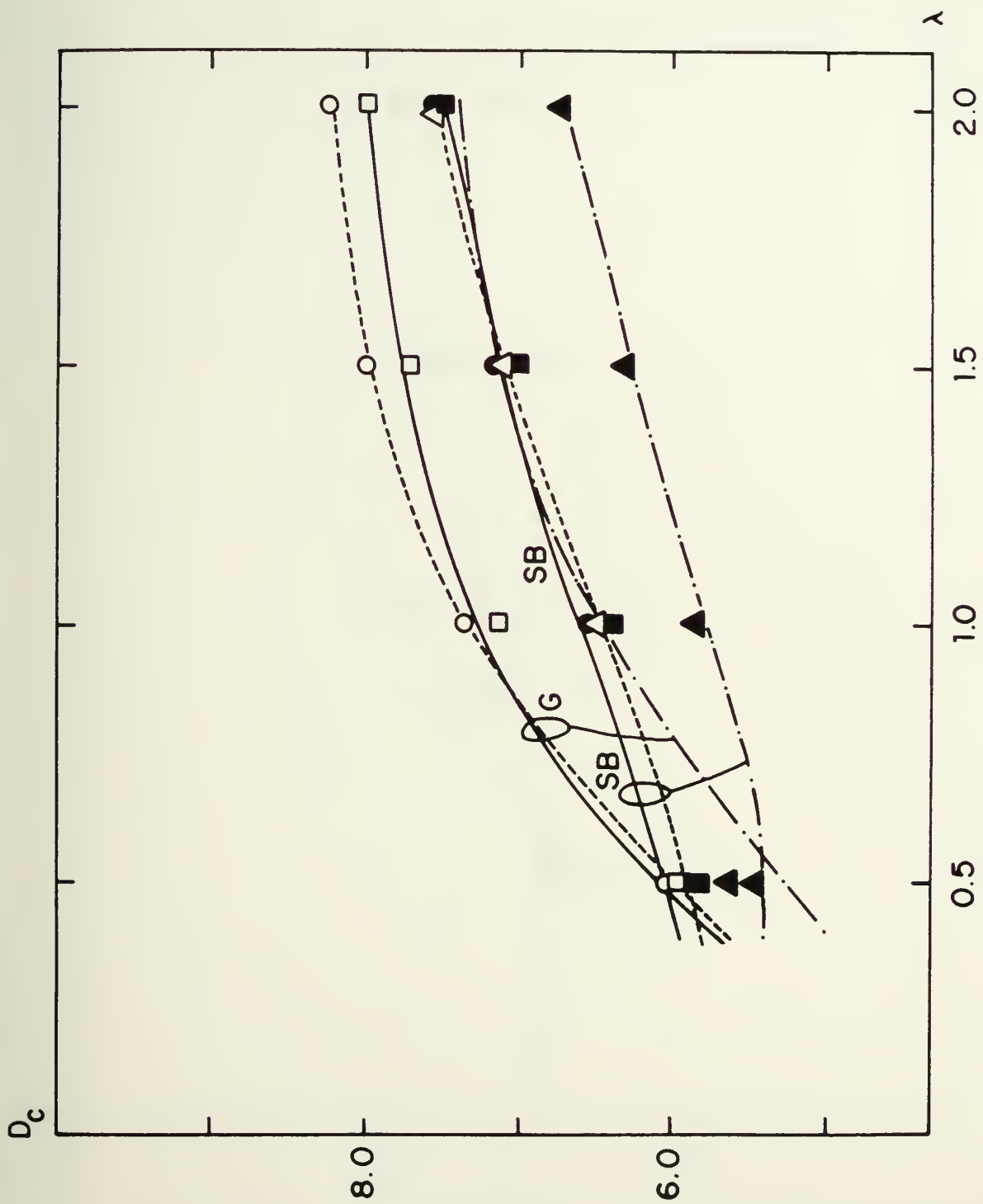
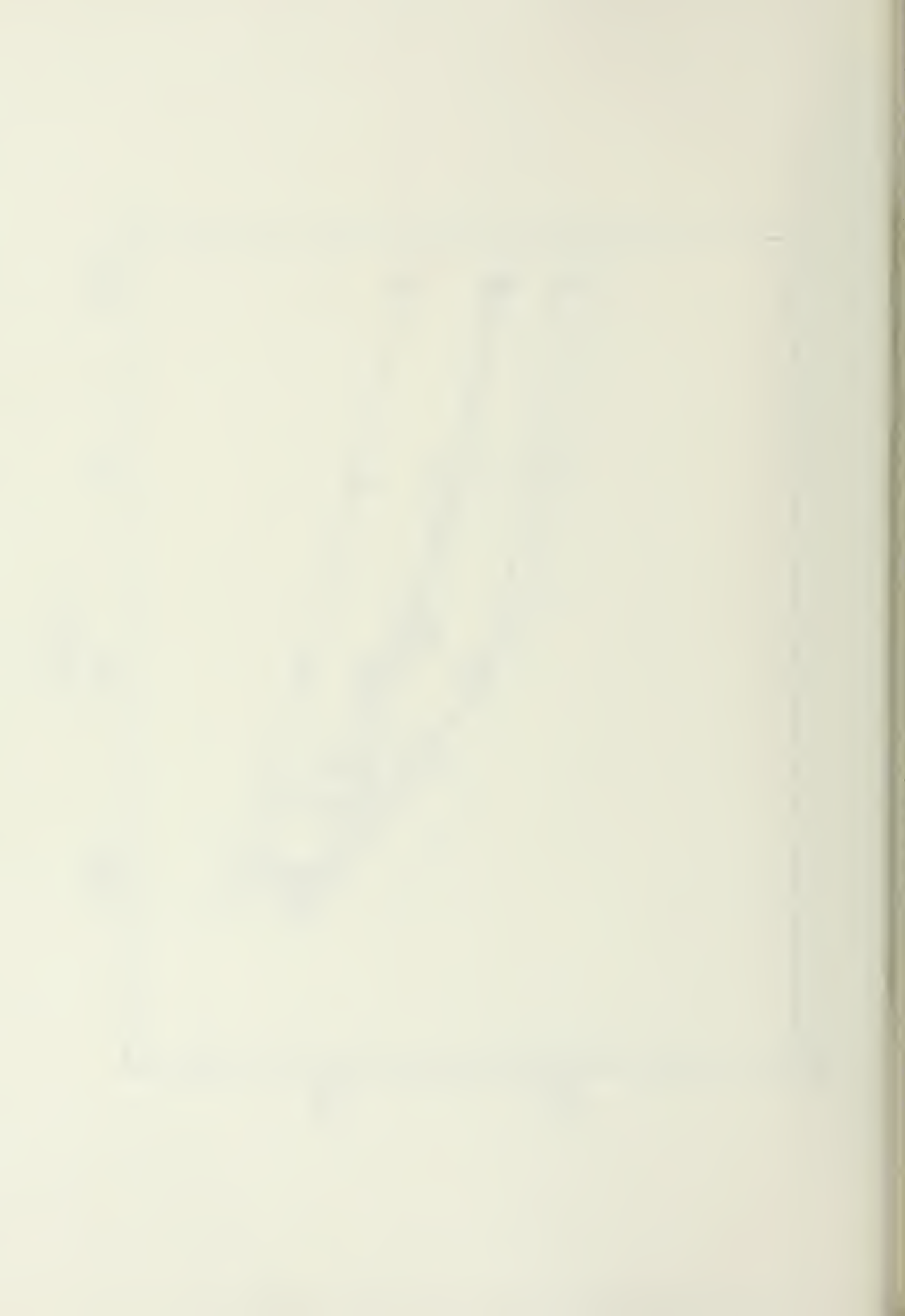


Figure 7.



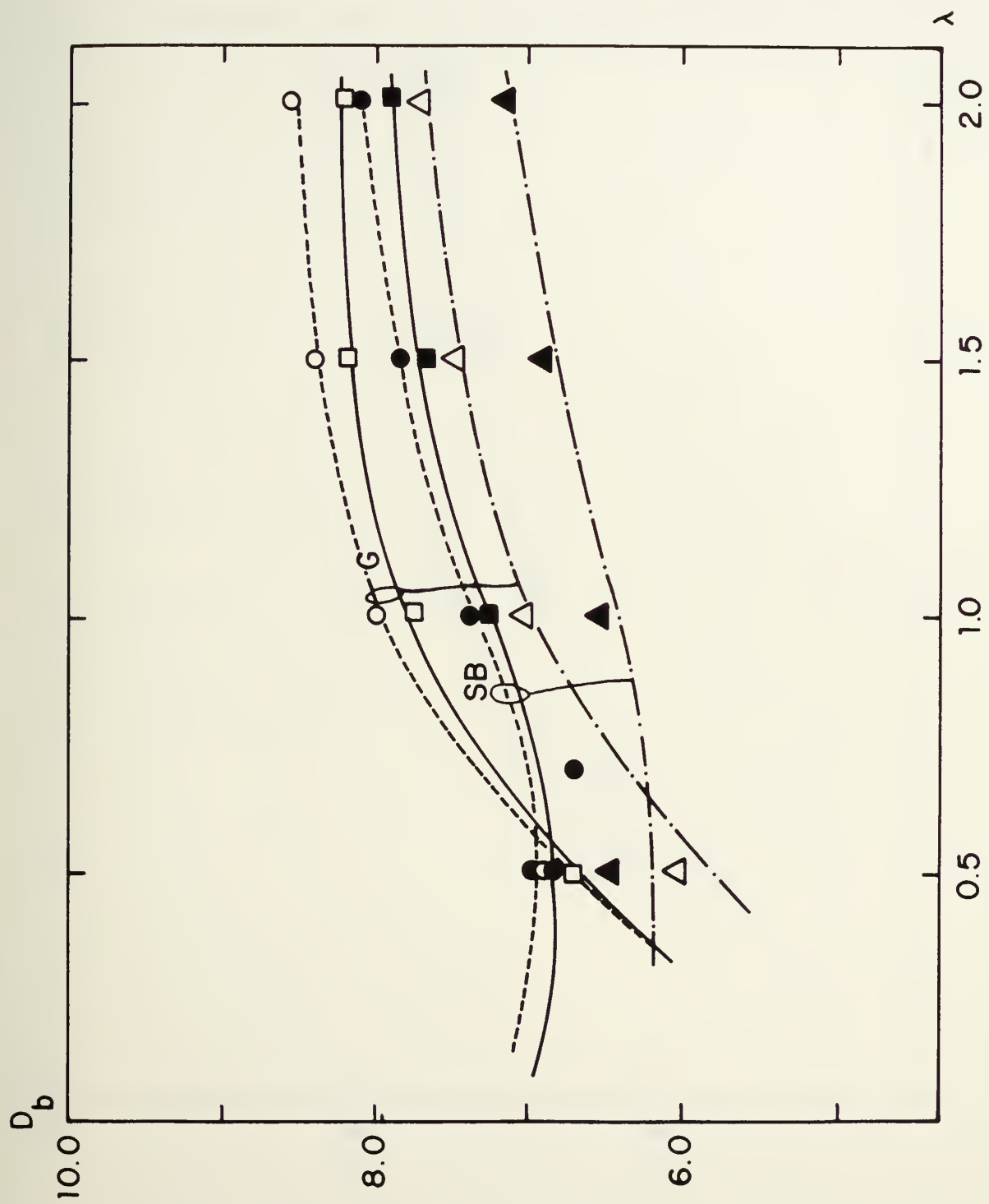


Figure 8.

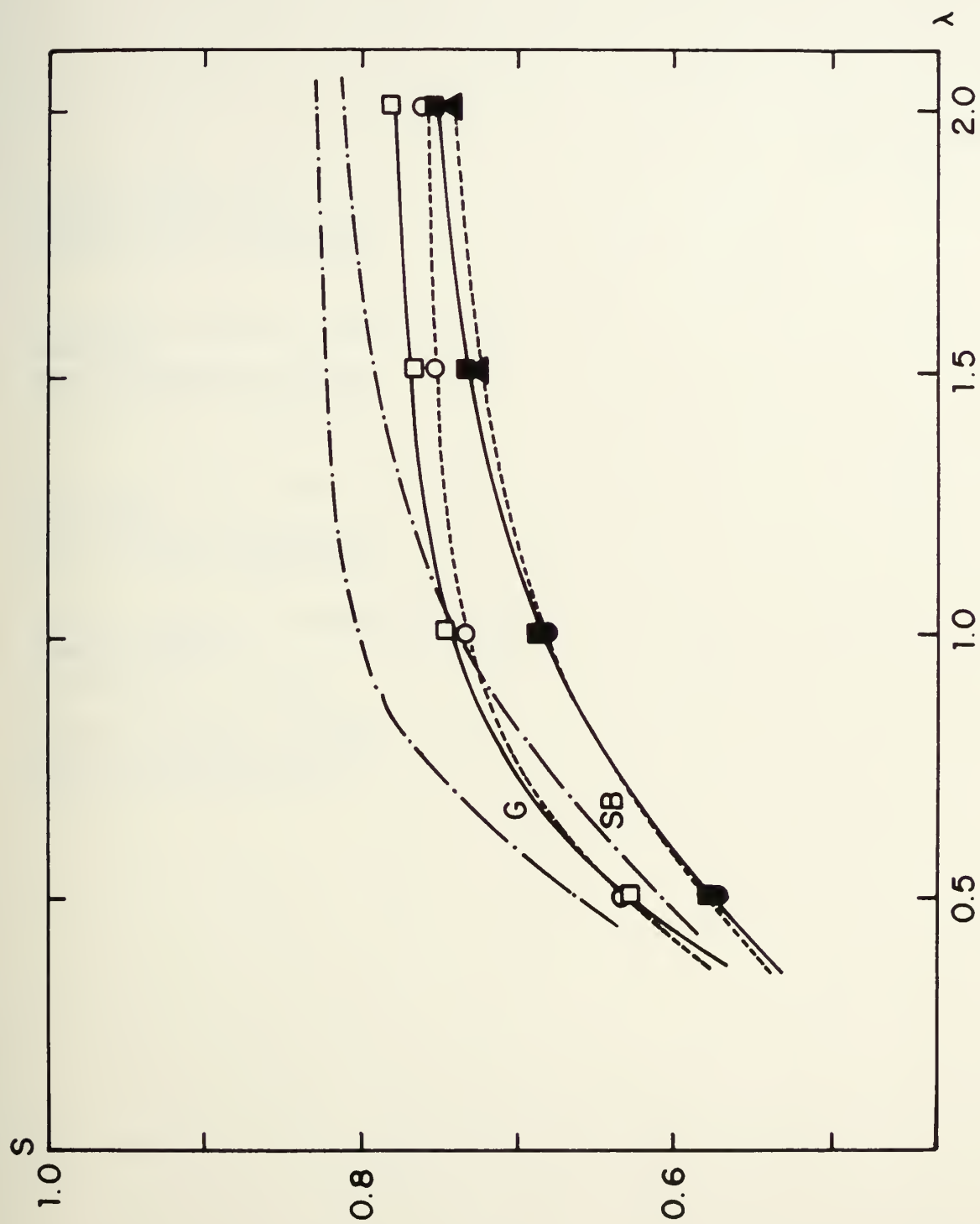
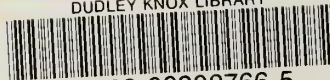


Figure 9.

DISTRIBUTION LIST

	NO. OF COPIES
Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5100	4
Director of Research Administration Code 012A Naval Postgraduate School Monterey, CA 93943-5100	1
Professor Jin-Fu Chang Code 62Cn Naval Postgraduate School Monterey, CA 93943-5100	5
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Center for Naval Analyses 2000 N. Beauregard Street Alexandria, VA 22311	1

DUDLEY KNOX LIBRARY



3 2768 00302766 5